Setup of Embedded Ethernet

This page: *How to set up ethernet.*

Introduction

Ethernet is the protocol used to communicate between all the components. It uses the **auto-generated Ethernet driver** code from cubeMX to use the physical Ethernet peripheral. It also uses **LWIP** to do the lowest levels of packet handling. Lastly, we use **freeRTOS** for multi threading. The current implementation cannot work without it.

CubeMX

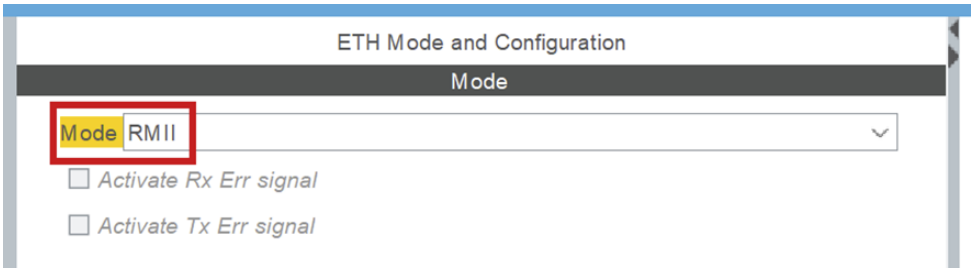
CubeMX is used to automatically generate setup code for the stm32. To use Ethernet you have to set up a few things in cubeMX.

To write this driver I mostly used videos from ControllersTech on YouTube. The most useful video is STM32 Ethernet (Part 1): How to configure Ethernet peripheral and perform successful ping test [1]. So if you do not understand anything, watch that video.

---

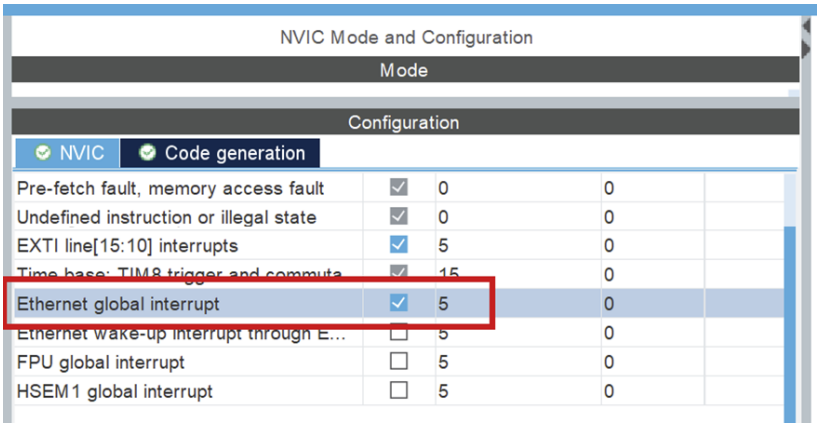
1) ETH

ETH is under connectivity in cubeMX. It sets up the Ethernet peripheral. Set it to `RMII` mode.



NVIC > NVIC

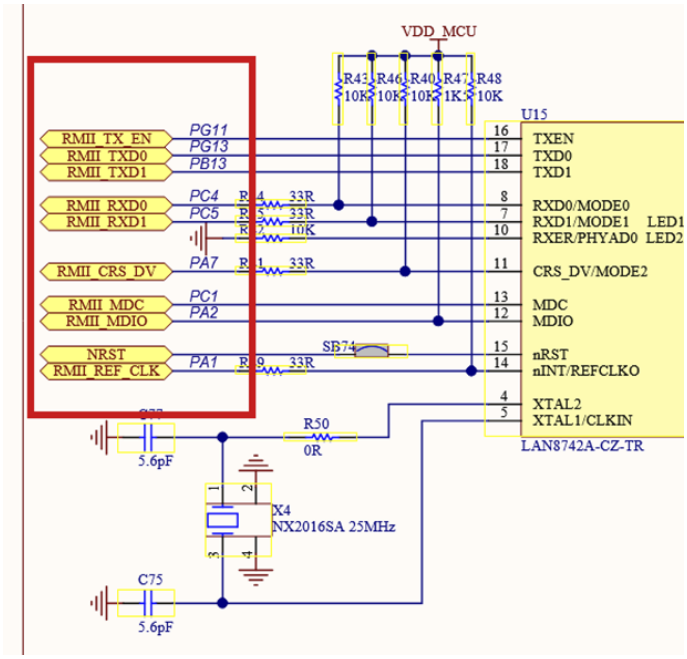
Turn on Ethernet global interrupt in NVIC settings.

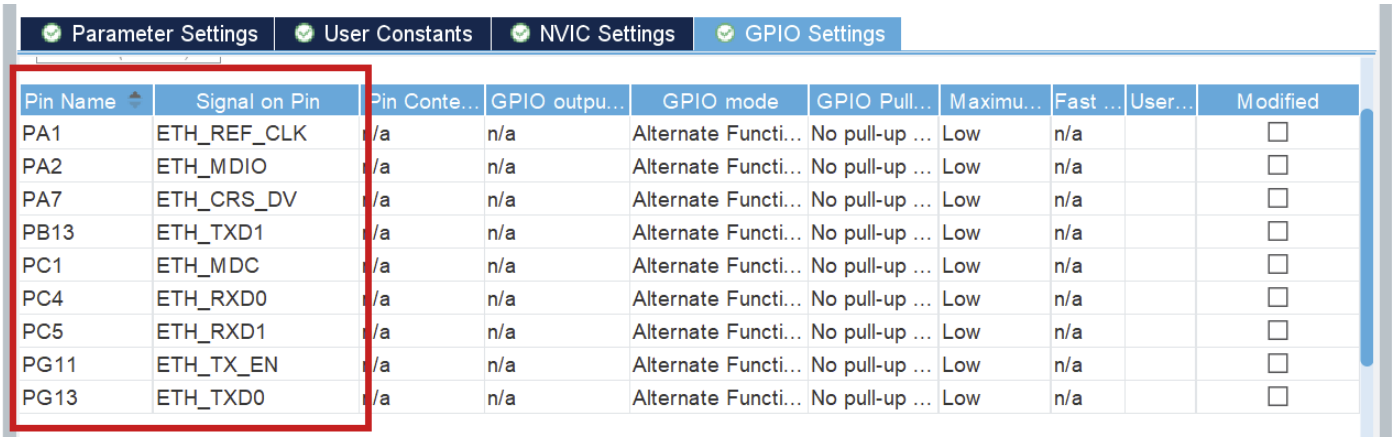


ETH > GPIO settings

> When Ethernet is enabled, the pins should **automatically** be configured according to the below schematic. This can be wrong, please **confirm it**!

Setup all the PINS like it is done in the PIN schematic (Download: MB1364-H753ZI-C01 Board schematic, pg. 6) [2].





Pin Name	Signal on Pin	Pin Conte...	GPIO outpu...	GPIO mode	GPIO Pull...	Maximu...	Fast ...	User...	Modified
PA1	ETH_REF_CLK	n/a	n/a	Alternate Functi...	No pull-up ...	Low	n/a		☐
PA2	ETH_MDIO	n/a	n/a	Alternate Functi...	No pull-up ...	Low	n/a		☐
PA7	ETH_CRS_DV	n/a	n/a	Alternate Functi...	No pull-up ...	Low	n/a		☐
PB13	ETH_TXD1	n/a	n/a	Alternate Functi...	No pull-up ...	Low	n/a		☐
PC1	ETH_MDC	n/a	n/a	Alternate Functi...	No pull-up ...	Low	n/a		☐
PC4	ETH_RXD0	n/a	n/a	Alternate Functi...	No pull-up ...	Low	n/a		☐
PC5	ETH_RXD1	n/a	n/a	Alternate Functi...	No pull-up ...	Low	n/a		☐
PG11	ETH_TX_EN	n/a	n/a	Alternate Functi...	No pull-up ...	Low	n/a		☐
PG13	ETH_TXD0	n/a	n/a	Alternate Functi...	No pull-up ...	Low	n/a		☐

Troubleshooting: Pins not set as schematic

> If any of the pins are not as in the schematic, refer to the above information. You can click each pin in CubeMX and choose the correct function (e.g. ETH_RXD0), the other (incorrect) pin will be automatically disabled.

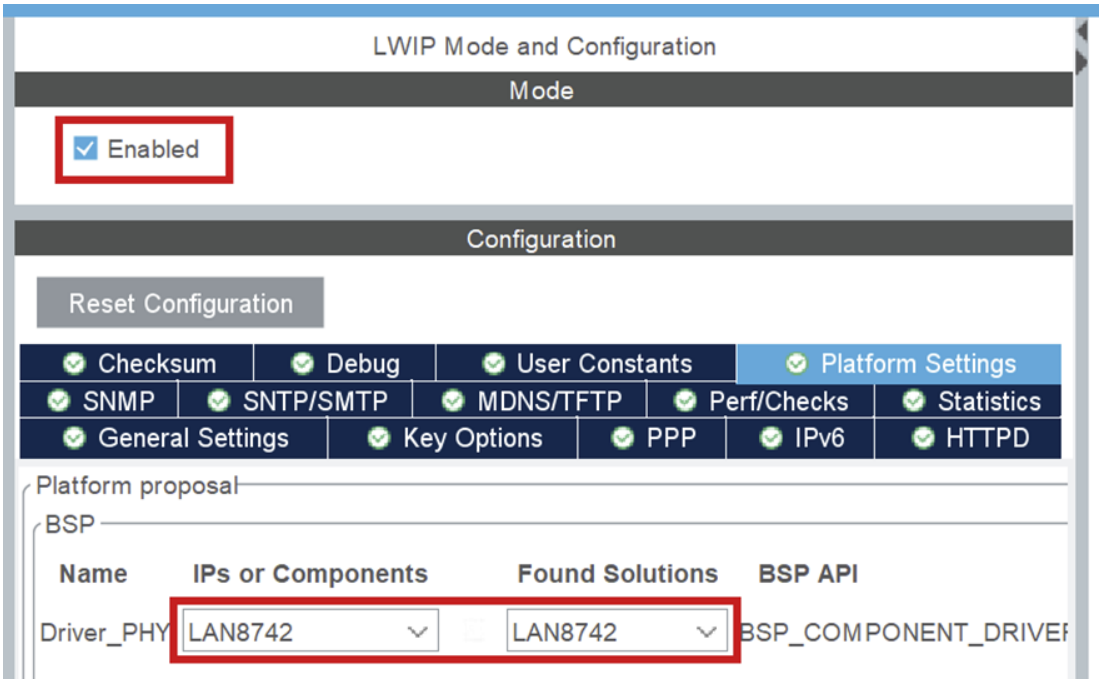
---

2) LWIP

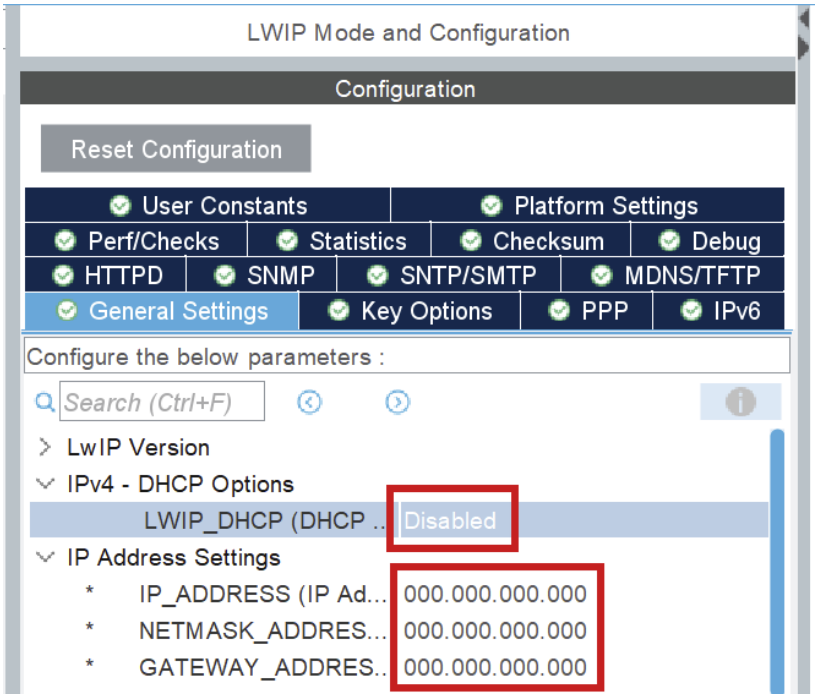
LWIP is a middleware generated by cubeMX. You can find it under middleware.

LWIP > Platform Settings

Set up LAN8742, to signal that that is the physical Ethernet driver you use.



LWIP > General settings

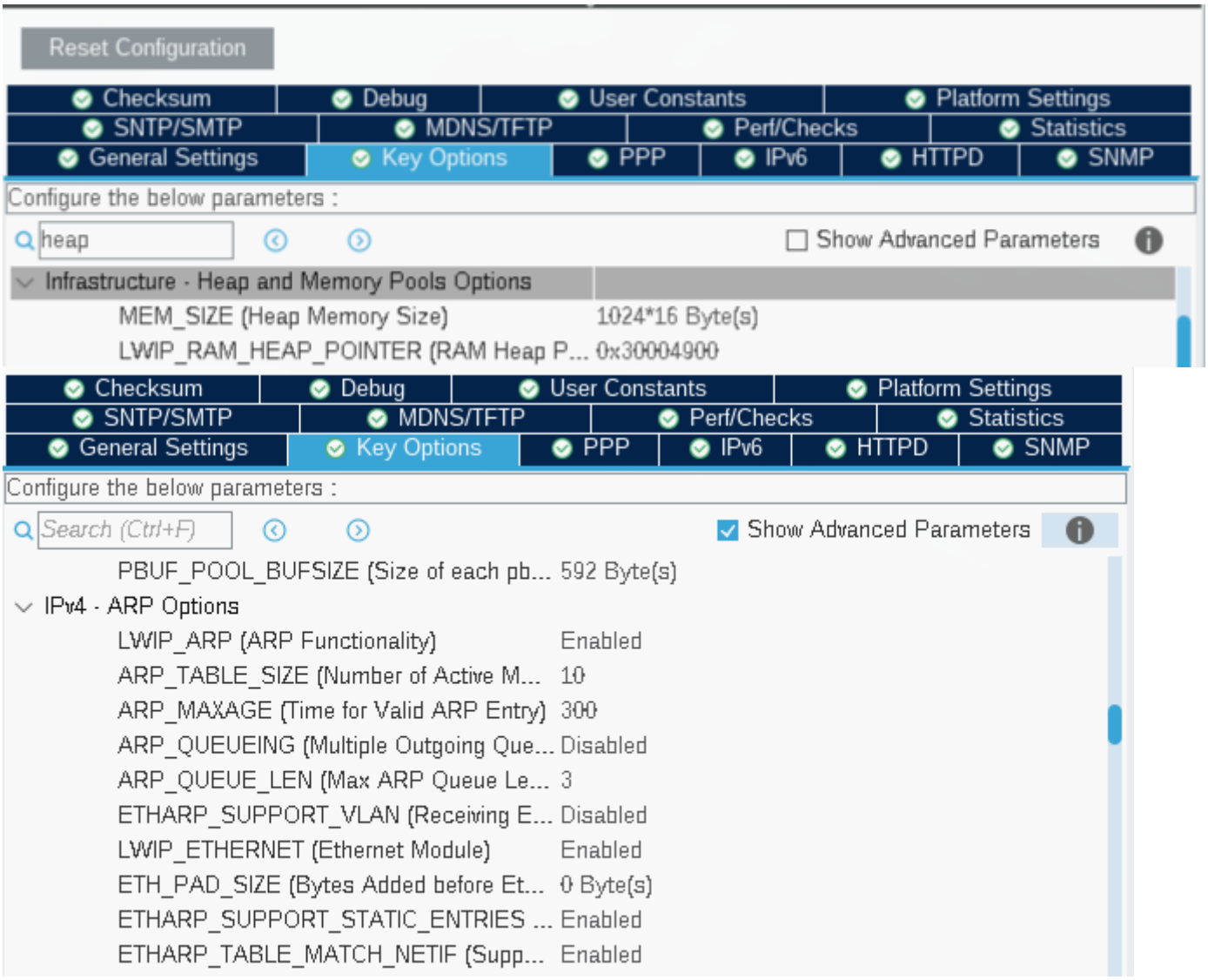


The most important configurations here are the DHCP (**DHCP = Disabled**) and IP address settings. The IP address settings **don't matter**, they will be **overwritten** in a configuration file in the code.

LWIP > Key options

The most important part of the tab *Key options* is to check if `LWIP_ARP = Enabled`. You also need to keep track of the `MEM_SIZE (Heap Memory Size)`. Start with a value of `1024*16 bytes`.

`LWIP_RAM_HEAP_POINTER` should be `0x30004900` such that the heap doesn't overlap different memory. That is the value I use, but it can be changed if need be.
Lastly, you need to add `ETHARP_SUPPORT_STATIC_ENTRIES = Enabled`. This is only shown when you turn on `Show Advanced Parameters`.



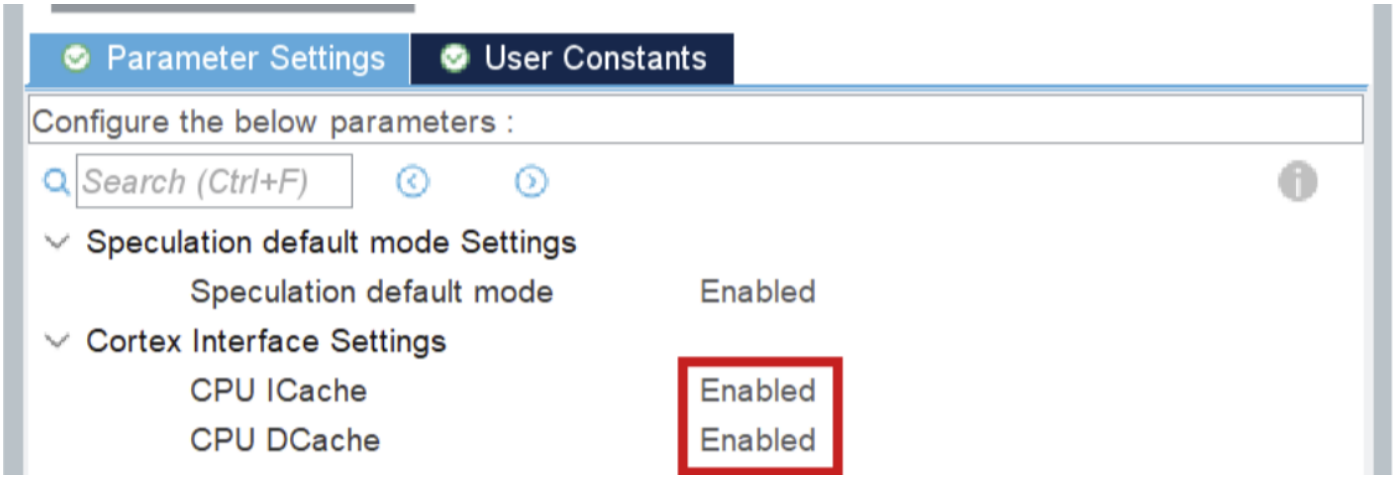
---

3) CORTEX_M7

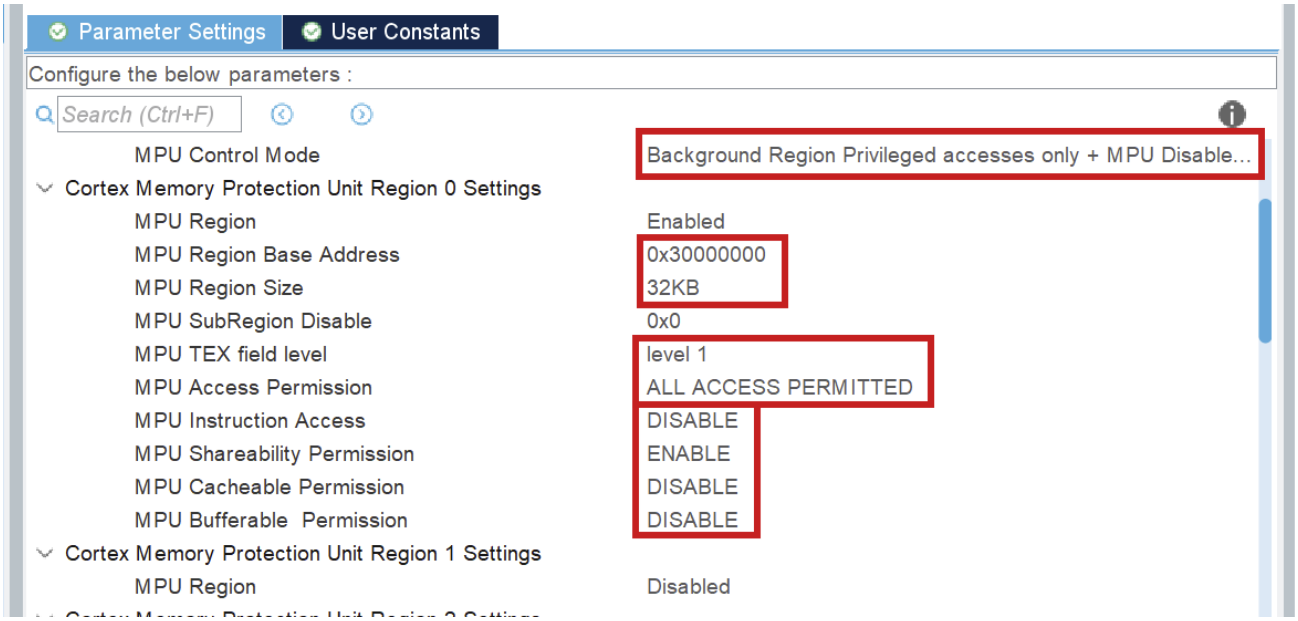
You can find *CORTEX_M7* under System Core.

CORTEX_M7 > Parameter Settings

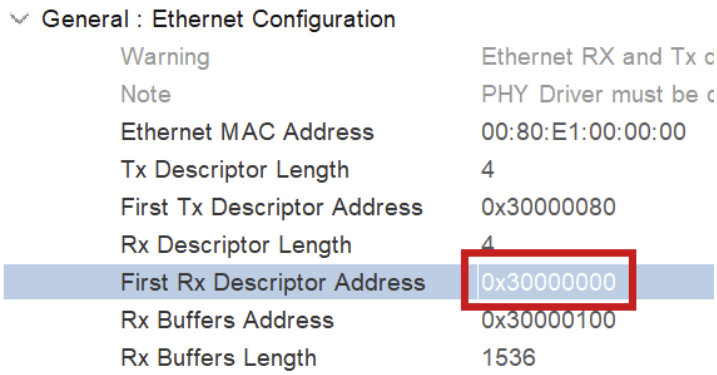
Turn on `CPU ICache` and `CPU DCache`.



Setup memory protection like seen below:

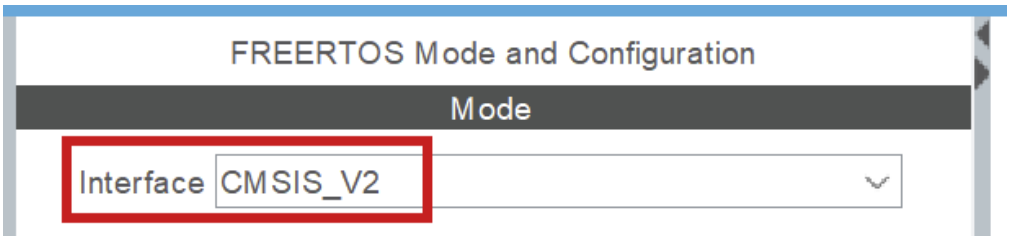


The `MPU Region Base Address` is the address of the first Rx descriptor (*ETH > Parameter Settings > General*). The `MPU region size` is calculated using heap memory size and the RX buffers. Watch the video [1] mentioned above for more information.



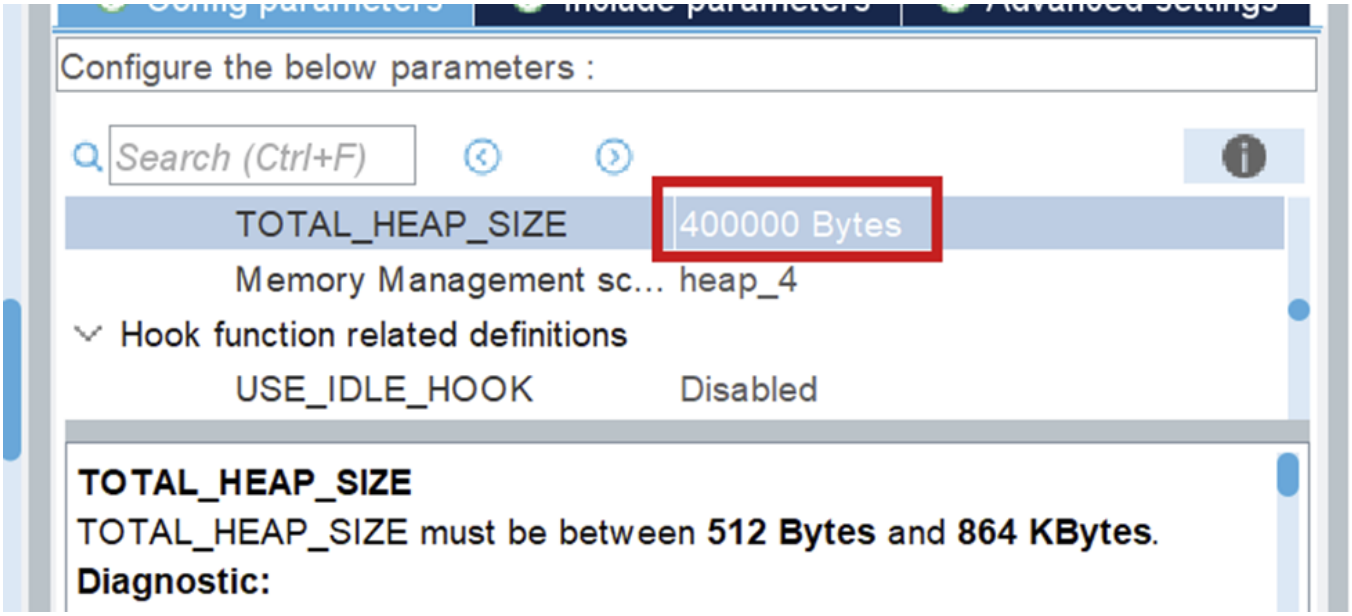
4) FreeRTOS

Freertos is automatically integrated in LWIP if you turn it on in cubeMX. It does not have to be modified, it just has to be turned on. Make sure you use `CMSIS_V2`, because V1 did not work well.



> **NOTE:** when you use FreeRTOS, you will have to select **another SYS Timebase Source**. Go to *SYS* > *Timebase Source* and select any of the free timers, just make sure it is **NOT** SysTick. Make sure you do not use that timer for any other activities.

Make sure you set your `TOTAL_HEAP_SIZE` to a sufficient number. When it is not big enough, the threads that you will create using FreeRTOS will suffocate and not work. This will also not give you any errors so watch out for it!



The Code

After you generate the code for your board, you can look through networking component, in the `ethernet.h` file, to see all public Ethernet functions.

> For a full guide of how to use Ethernet, I refer you to Driver usage [3].

Resources

1. STM32 Ethernet (Part 1): How to configure Ethernet peripheral and perform successful ping test
2. PIN schematic (Download: MB1364-H753ZI-C01 Board schematic, pg. 6)
3. Driver usage

Revision #31
Created 2026-04-16 13:42:17 UTC by Jaron Lendering
Updated 2026-06-12 13:41:40 UTC by Lisa te Braak